



Why Validate Form Inputs?

In real-world applications like **user registration**, **login**, or **checkout forms**, validating inputs ensures:

- Data is in the correct format
- Required fields are filled
- Prevents malicious data (e.g., SQL injection, XSS)
- Improves user experience by catching errors early

Popular Node.js Validation Libraries

1. **Joi** – Schema-based validation (widely used)
2. **express-validator** – Middleware for express using validator.js
3. **yup** – Similar to Joi but works great with frontend and backend
4. **validator.js** – Lightweight string validation library

We'll focus on **Joi** and **express-validator**, as they are best suited for server-side form validation.

Installation

```
npm install joi
```

Example: User Registration Form

Requirements:

- username: required, alphanumeric, min 3, max 30
- email: required, valid email
- password: required, min 8 characters

Real World Use Case:

This could be used in a user sign-up API for an e-commerce platform.

```
const Joi = require('joi');

// Define the schema

const schema = Joi.object({

  username: Joi.string().alphanum().min(3).max(30).required(),

  email: Joi.string().email().required(),

  password: Joi.string().min(8).required()

});

// Sample input

const formData = {

  username: "JensBett",

  email: "jens@myemail.com",
```

```
password: "mypassword123"

};

// Validate

const { error, value } = schema.validate(formData);

if (error) {

  console.log("Validation Error:", error.details[0].message);

} else {

  console.log("Validated Data:", value);

}
```

Using **express-validator** with Express

npm install express-validator

Example: Contact Form

Requirements:

- name: required
- email: valid email
- message: required, max 500 chars

```
const express = require('express');

const { body, validationResult } = require('express-validator');
```

```
const app = express();

app.use(express.json());

app.post('/contact', [

  body('name').notEmpty().withMessage('Name is required'),

  body('email').isEmail().withMessage('Must be a valid email'),

  body('message').isLength({ max: 500 }).withMessage('Message too long')

], (req, res) => {

  const errors = validationResult(req);

  if (!errors.isEmpty()) {

    return res.status(400).json({ errors: errors.array() });

  }

  // Proceed to save the message

  res.status(200).send("Message received!");

});

app.listen(3000, () => console.log('Server running on port 3000'));
```

